Metric Correlation and Analysis Service (MCAS)

The MCAS mission is to deliver the software product to help with adaptation, retrieval, correlation and display of type agnostic and disjoint middleware generated events and workflow driven data.

Mission need

It's common that distributed software stack or grid workflow can span organization, ownership and deployment domains. In this setting, such important and common tasks as metric and debug information collection, display and ultimately problem troubleshooting are challenged by informational entropy inherent to independently maintained and operated software sub components. Because such an information "pond" is disorganized, it becomes a difficult target for business intelligence analysis i.e. troubleshooting, incident investigation and trend spotting.

Project goal.
Our goal is to factor out presentation and business analysis logic from available monitoring solutions into a standalone model supporting common standard in means for data manipulation and presentation. We have prototyped several services which relay on common techniques for data and information display aggregation. In particular , we've chosen portlet technology to compose troubleshooting and metric analysis dashboard and ESB Mule to drive data integration model. We have used portlet JSR128 specification and Jboss engine for display integration and Mule ESB to transform external data for consumption by the portlet code.

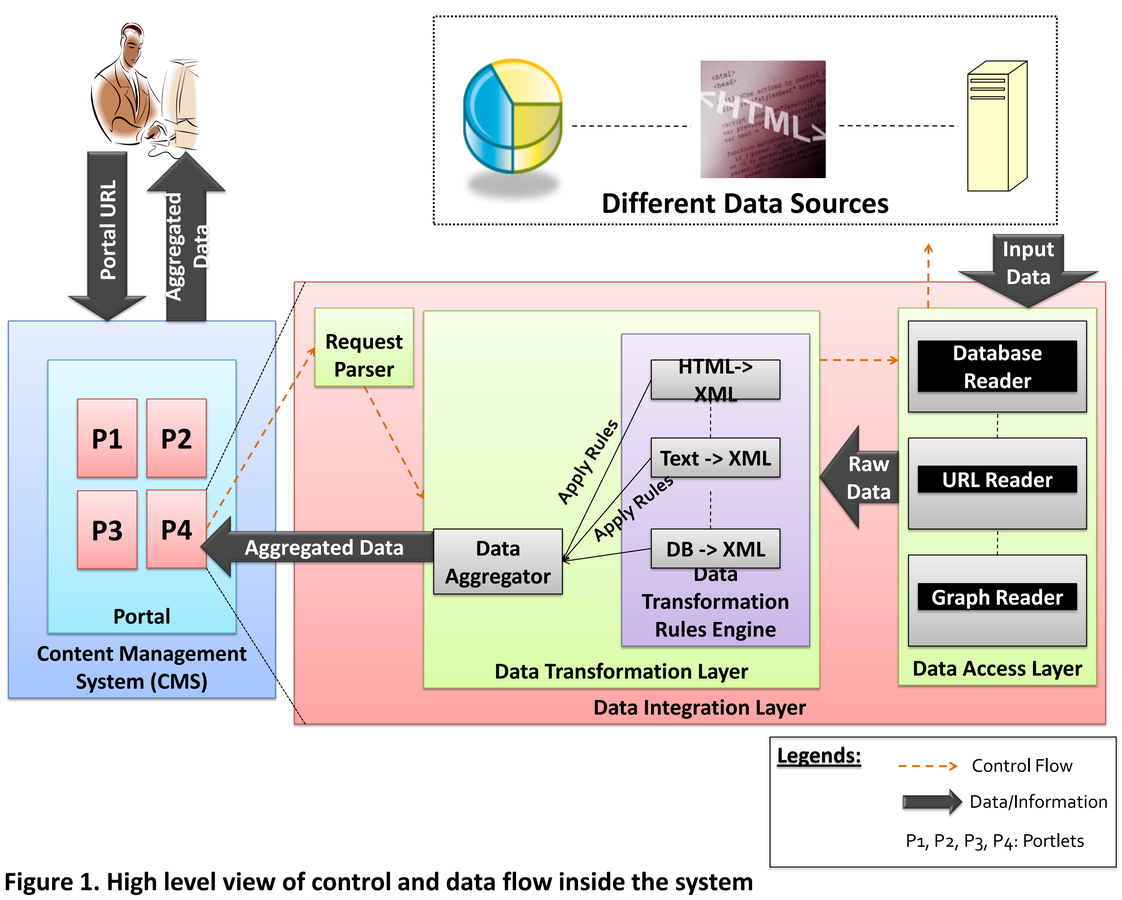

Figure 1. High level view of control and data flow inside the system

Workflow

JBoss portal content management system builds composition of independent interface elements – portlets (P1 , P2, P3, P4). Each portlet can be independently designed with some unique perspective of the particular system aspects. The composition of such elements is a dashboard of indicators specifically put together to comprehensively reflect the state of the entire system.

Each portlet is rendered using information provided by the data integration layer.

Data integration layer accesses set of data sources and uses collection of rules to transform and aggregate the retrieved content. Data integration layer generates digest content with only those details that are relevant to rendering of the portlet itself.

The result is returned synchronously to a requesting party, which may be portlet or parent data integration rule set.

Summary

Data integration layer defines all url endpoints providing content for the display. These endpoints may be as simple as proxies to existing web pages or hide rules for transforming and aggregating data retrieved from other sources. The purpose of this complexity is to provide data that is not unavailable in content preferred by the user interface.

Data integration model

Data integration model relies on Mule ESB for data source access , inter component message exchange and data transformation scheduling. The primary benefit of Mule ESB integration platform is in its features to manage data and execution flow in a transport/interface agnostic way. In particular Mule ESB offers:

- Codes to translate or templatise translation of data formats
- Options of manage synchronicity with choice ranging from fully synchronous to SEDA (stage event driven architecture) based solutions.
- Codes which adapt out of the box to different transports (TCP, UDP, SMTP, FTP, jdbc, etc)

Messaging

Message exchange is a clever way to decouple contexts of two programs. Messaging is a soft pattern which does not imply fixed db schema or file format behind. It can encapsulates data transport and synchronicity issues. Most importantly using Mule message enabled infrastructure we can use opaque payloads and do modeling of the data access and aggregation work flow w/o setting the specifics of the data source type structures.

Phase I of the MCAS project puts together infrastructure of tools and services for efficient refactoring of existing informational portals with capabilities of aggregating and correlating information provided by those portals.

Figure 2 depicts one of the implemented scenarios which uses data transformation workflow and RRD processing engine to do splitting, rescaling and redrawing of D0SiteEfficiency data by the RRD processing engine.

Transformation pipeline: Example

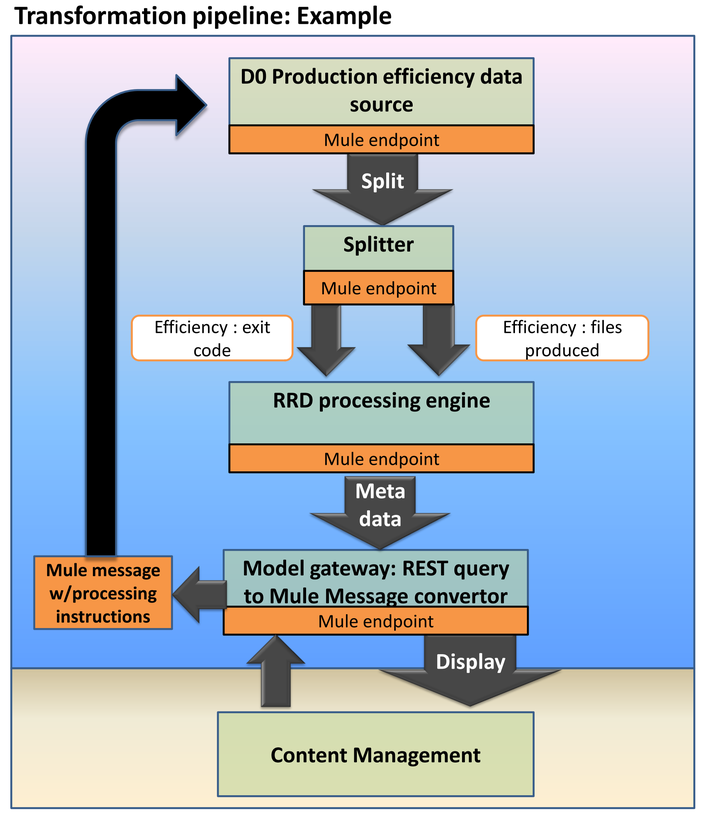

Figure 2

Data transformation engine is using "models" to drive the message interactions between Mule ESB message endpoints. This particular above schema is designed to understand template like language and has only one data source (production efficiency) endpoint. The embedded RRD template allows to perform transformations over split data streams . The result of the transformation -new document (an image) is sent to the portlet instance specifically configured to interact with this data integration model.

Data source transformation template language: Example

```
ds(D0ProductionEfficiency)
   ec=eff_code; ef=eff_fini; RRD( CDEF:ec_adj=ec,0,100,LIMIT CDEF:ef_adj=ef,0,100,LIMIT
LINE2:ec_adj#FF0000:eff_code(x100) LINE2:ef_adj#0000FF:eff_fini(x100) )  imgsize(600,300)
```

Work supported by the U.S. Department of Energy under contract No. DE-AC02-07CH11359